EL9343 Homework 1

(Due September 21th, 2021)

No late assignments accepted

All problem/exercise numbers are for the third edition of CLRS text book

---

1. Prove the *Transpose Symmetry* property, i.e. f(n) = O(g(n)) if and only if g(n) = Ω(f(n))

2. Problem 3-1 in CLRS Text book.

3. Problem 3-2 in CLRS Text book.

4. You have 5 algorithms, A1 took $O(n)$ steps, A2 took $\Theta(n \log n)$ steps, and A3 took $\Omega(n)$ steps, A4 took $O(n^3)$ steps, A5 took $o(n^2)$ steps. You had been given the exact running time of each algorithm, but unfortunately you lost the record. In your messy desk you found the following formulas:

(a) $3n\log_2 n + \log_2 \log_2 n$

(b) $3(2^{2\log_2 n}) + 5n + 1234567$

(c) $\frac{2^{\log_4 n}}{3} + n + 9$

(d) $(\log_2 n)^2 + 5$

(e) $3n!$

(f) $2^{3\log_2 n}$

(g) $2^{2\log_2 n}$

For each algorithm write down all the possible formulas that could be associated with it.

5. For the following algorithm: Show what is printed by the following algorithm when called with MAXIMUM(A, 1, 5) where A = [10, 8, 6, 4, 2]? Where the function PRINT simple prints its arguments in some appropriate manner.

```
MAXIMUM(A, l, r)
if (r − l == 0)
    return A[r]

lmax = MAXIMUM(A, l, ⌊(l + r)/2⌋)
rmax = MAXIMUM(A, ⌊(l + r)/2⌋ + 1, r)
PRINT(rmax, lmax)
if rmax < lmax
    return lmax
else
    return rmax
```

6. First use the iteration method to solve the recurrence, draw the recursion tree to analyze.

$$T(n) = T(\frac{n}{2}) + T(\frac{n}{3}) + n$$

Then use the substitution method to verify your solution.

7. Use the substitution method to prove that $T(n) = 2T(\frac{n}{2}) + cnlog_2 n$ is $O(n(log_2 n)^2)$.

8. Solving the recurrence:

$T(n) = 3T(\sqrt[3]{n}) + log_2 n$

(Hint: Making change of variable)

9. You have three algorithms to a problem and you do not know their efficiency, but fortunately, you find the recurrence formulas for each solution, which are shown as follows:

A: $T(n) = 3T(\frac{n}{3}) + \theta(n)$

B: $T(n) = 2T(\frac{9n}{10}) + \theta(n)$

C: $T(n) = 3T(\frac{n}{3}) + \theta(n^2)$

Please give the running time of each algorithm (In θ notation), and which of your algorithms is the fastest (You probably can do this without a calculator)?

10. Can the master theorem be applied to recurrence of $T(n) = T(\frac{n}{2}) + n^2 lgn$? Why does it work or not? Provide the asymptotic upper bound for this recurrence.